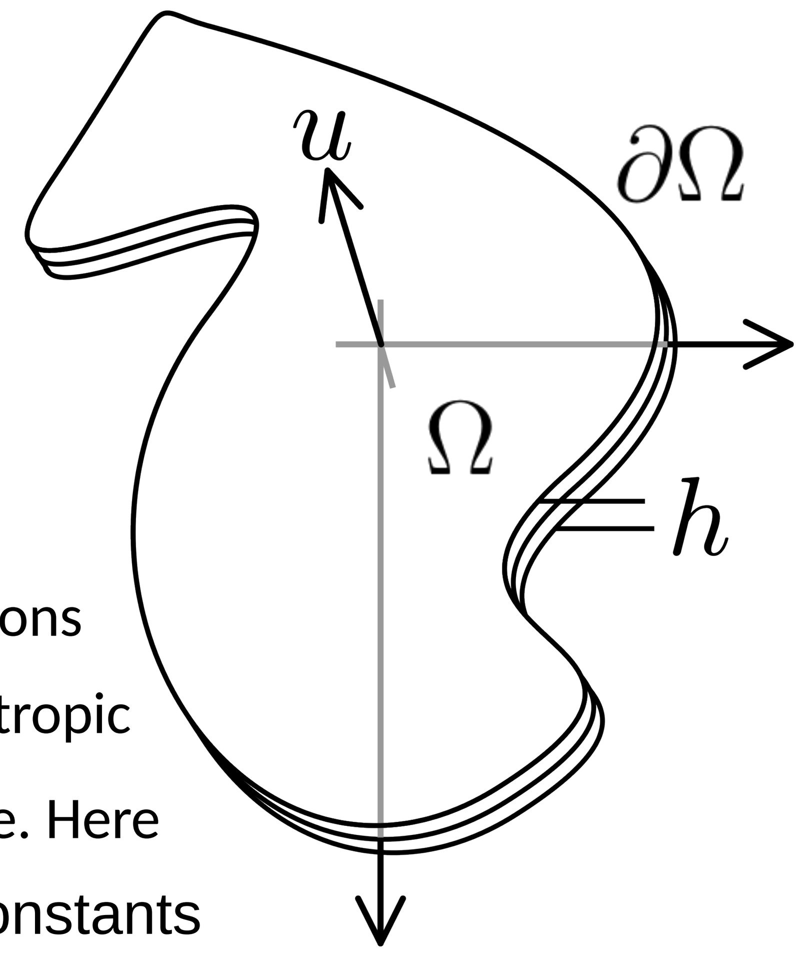
Eigenfunctions of the Plate Equation | Alexander Jansing, Aaron Gregory

Faculty Advisor: Dr. Andrea Dziubek

SUNY POLYTECHNIC INSTITUTE
COLLEGE OF ARTS + SCIENCES



MODEL

We begin with the determining equations for a supported isotropic Kirchhoff-Love plate. Here K, ρ, and h are constants reflecting physical attributes of the modeled plate.

$$\frac{\rho h}{K}\ddot{w} + \Delta^2 w = 0, \qquad \forall \boldsymbol{x} \in \Omega, t > 0$$
$$u = \Delta u = 0, \qquad \forall \boldsymbol{x} \in \partial\Omega$$

We're interested in vibration modes of our plate, and we do separation of variables accordingly:

$$w = \cos(\omega t) u(\boldsymbol{x})$$

We then divide by $\cos(\omega t)$ and substitute $\frac{\rho h \omega^2}{K} = \lambda^2$. The resulting equation is

$$\Delta^2 u = \lambda^2 u, \qquad \forall \boldsymbol{x} \in \Omega$$

After performing reduction of order we find the matrix form of our eigenvalue problem

$$\begin{bmatrix} 0 & \Delta \\ \Delta & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \end{bmatrix}$$

DISCRETIZATION

To expand our potential function space from C^2 to C^1 we integrate both sides of our matrix equation after multiplying by test functions:

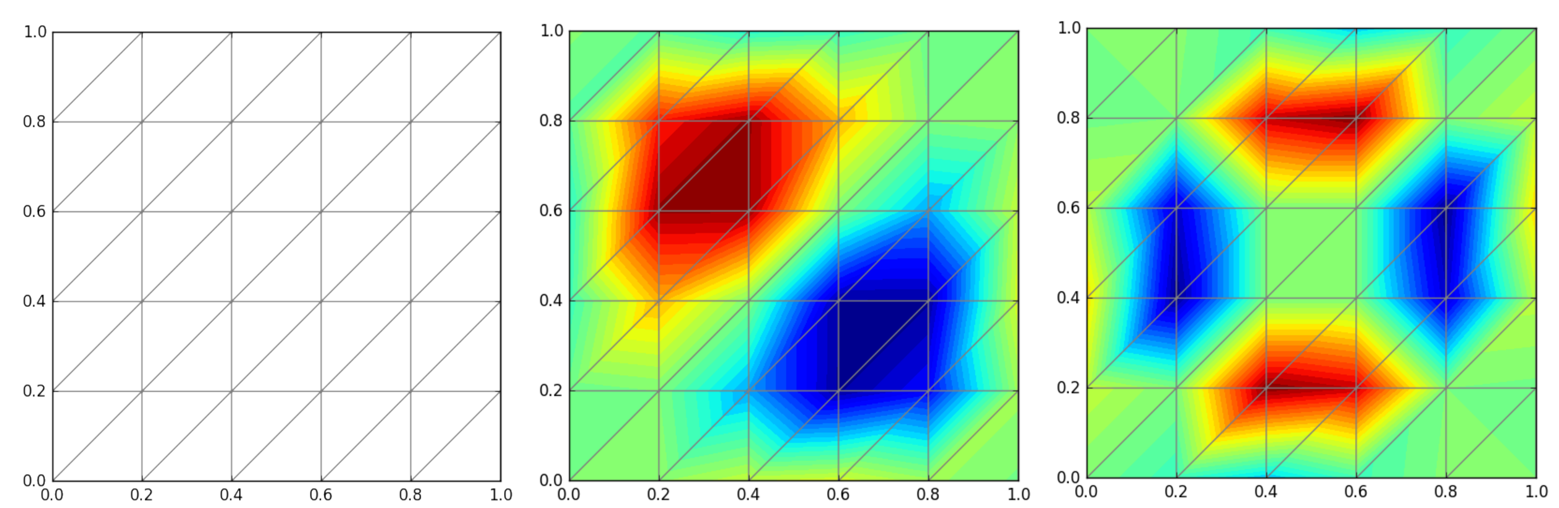
$$\int_\Omega \begin{bmatrix} \phi \\ \varphi \end{bmatrix}^T \begin{bmatrix} 0 & \Delta \\ \Delta & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} d\boldsymbol{x} = \int_\Omega \lambda \begin{bmatrix} \phi \\ \varphi \end{bmatrix}^T \begin{bmatrix} u \\ v \end{bmatrix} d\boldsymbol{x}$$

After integrating by parts and negating, we get our weak formulation, which we will solve by discretization across a mesh.

$$-\int_\Omega \nabla v \cdot \nabla \phi + \nabla u \cdot \nabla \varphi \; d\boldsymbol{x} = \int_\Omega \lambda u \phi + \lambda v \varphi \; d\boldsymbol{x}$$

FINITE ELEMENT METHOD

We use the method of finite elements to solve our weak formulation by taking $\phi, \varphi \in \mathcal{B}$, where $\mathcal{B}$ consists of piecewise-linear functions defined on the nodes of our mesh. u and v will be members of the function space spanned by $\mathcal{B}$, so the density of our mesh corresponds directly to the resolution of our solutions.



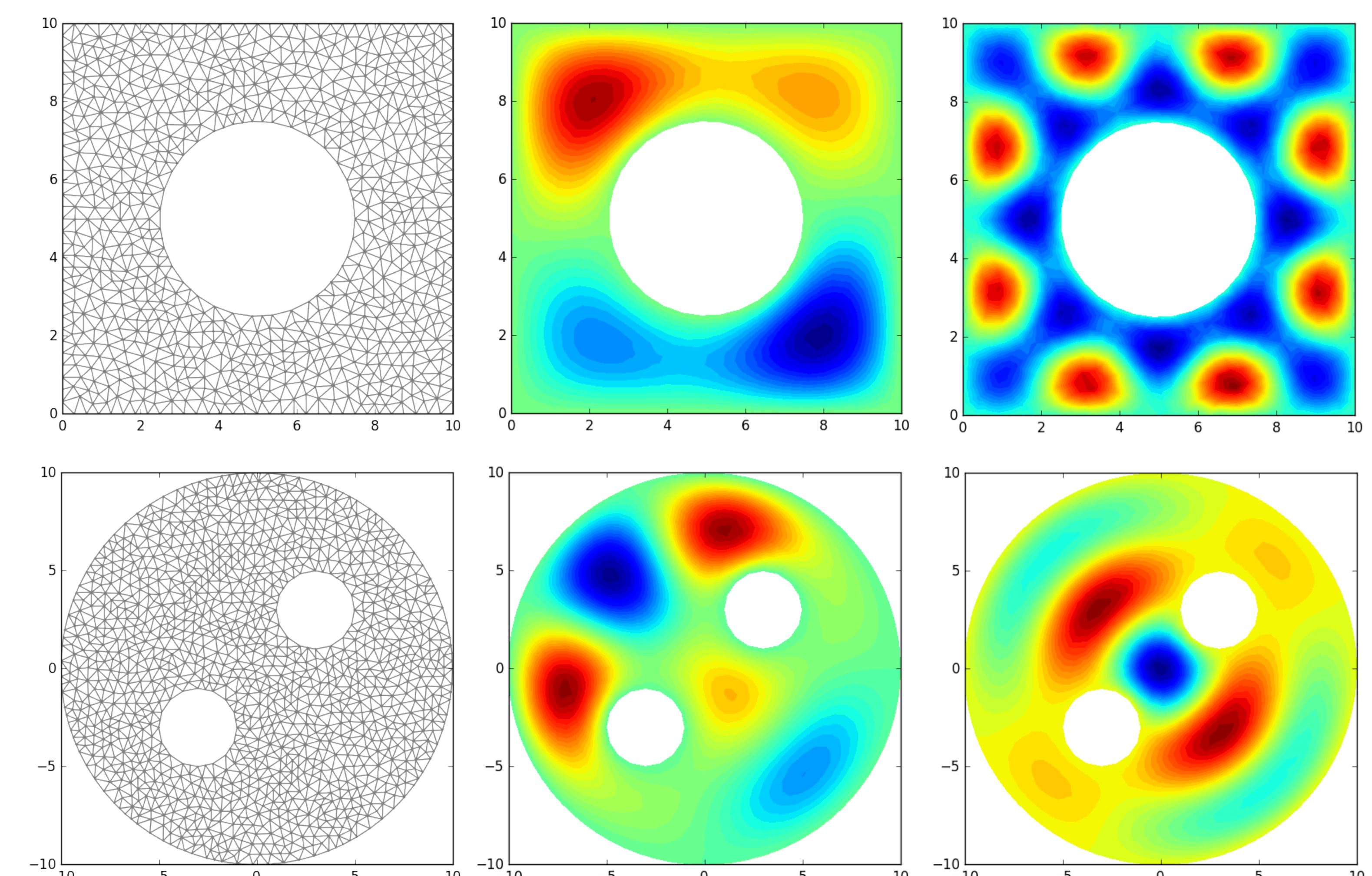
COMPUTATION

We use FEniCS for our numerical work. A matrix is generated from our weak formulation and mesh, and we then solve for its eigenvalues and eigenvectors, which represent the patterns of vibration possible on our plate. Surprisingly little code is needed to implement this problem in FEniCS.

```
class DirichletBoundary(SubDomain):
    def inside(self, x, on_boundary):
        return on_boundary

# Define boundary condition
u0 = Constant(0.0)
bc1 = DirichletBC(W.sub(0), u0, DirichletBoundary())
bc2 = DirichletBC(W.sub(1), u0, DirichletBoundary())

# Define the bilinear form
(u, v) = TrialFunction(W)
(f1, f2) = TestFunction(W)
a = -(dot(grad(u), grad(f2)) + dot(grad(v), grad(f1)))*dx
L = (u*f1 + v*f2)*dx

# Create the matrices
A = PETScMatrix()
b = PETScMatrix()
assemble(a, tensor = A)
assemble(L, tensor = b)
bc1.apply(A)
bc2.apply(A)

# Create eigensolver
eigensolver = SLEPcEigenSolver(A, b)

# Compute all eigenvalues of A x = \lambda x
print("Computing eigenvalues. This can take a minute.")
eigensolver.solve()
```

RESULTS

Unsurprisingly, we found that the number of converging eigenfunctions increases in a manner proportionate to the size of the mesh. We found 71 eigenfunctions for our small square mesh, and 2425 for our large circular mesh. Some of these were duplicates created due to rounding errors, although from visual inspection it appears that almost all are unique.



REFERENCES

1. M. S. Alnaes et. al. *The FEniCS Project Version 1.5*, Archive of Numerical Software, vol. 3, 2015
2. A. Logg and G. N. Wells *DOLFIN: Automated Finite Element Computing*, ACM Transactions on Mathematical Software, vol. 37, 2010
3. C. Geuzaine and J.-F. Remacle. *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post- processing facilities*. International Journal for Numerical Methods in Engineering 79(11), pp. 1309-1331, 2009